Modeling Military Deployment in Theaters of Operations – Balancing Deployment Alternatives

Charles N. Van Groningen, Ph.D., Dariusz Blachowicz, Mary Duffy Braun, Richard J. Love, Kathy Lee Simunich, and Mary Ann Widing,

Abstract: This paper describes how military analysts can use ELIST, the Enhanced Logistics Intra-theater Support Tool, to evaluate the impact of a wide range of conflicting priorities while developing plans for the deployment of forces into theaters of operation. These priorities include options such as delaying transportation assets to allow more security forces in early, delaying security forces for port offload forces, substituting smaller and cheaper vehicles for those with greater capabilities, and varying command and control configurations. The data requirements and management techniques of ELIST are described.

Keywords: military deployment simulation alternatives

1 Introduction

ELIST, the Enhanced Logistics Intra-theater Support Tool, is a transportation forecasting simulation that models the movements of troops, equipment and re-supply cargo from ports of debarkation or theater origins, to final destinations or tactical assembly areas [1]. It simulates organic and external lift movements over a detailed model of the infrastructure with constrained transportation assets [2][3]. Six types of data are required to perform analyses:

- Vehicle characteristics for each type of vehicle including: trucks, railcars, helicopters, fixed-wing aircraft, and watercraft [4][5]
- A detailed network representation characterizing the geospatial representation of the road, rail, waterway, and pipeline infrastructure; as well as the capacity of each location to process cargo
- A set of troops, equipment, and supplies and their required movements through the theater. These movements include where the units originates and when they are available to move; their destinations, what transport modes (e.g. road) they may use, and when they must be there. Cargo descriptions vary in detail from generic short-tons of supply to specific pieces of equipment.
- A set of rules that characterize the cargo into "commodities" (user defined cargo types) and provide prioritized options for which vehicles can move each commodity
- A set of scenario options for altering the assumptions of the run
- A set of specific transport vehicles that can move the cargo through the network

The plans evaluated by ELIST can be very large encompassing moving over 80,000 units over a network with over 10,000 links and nodes using a fleet of over 1,000 vehicles. The ELIST GUI and simulation is written in Java and interacts with an Oracle[tm] database to store these very large data sets.

This paper begins by describing some technical concepts used to store and organize ELIST data. Section 3 describes how users can configure scenarios for analyzing alternatives. Section 4 provides examples of how alternatives can be evaluated. Lastly some conclusions are presented.

2 Technical Concepts

This section describes three concepts used in ELIST to help manage the large quantity of data required to do an analysis. The context manager presents data to the user, the change log manages changes to the data, and the persistence layer stores the data and results (see Figure 1).

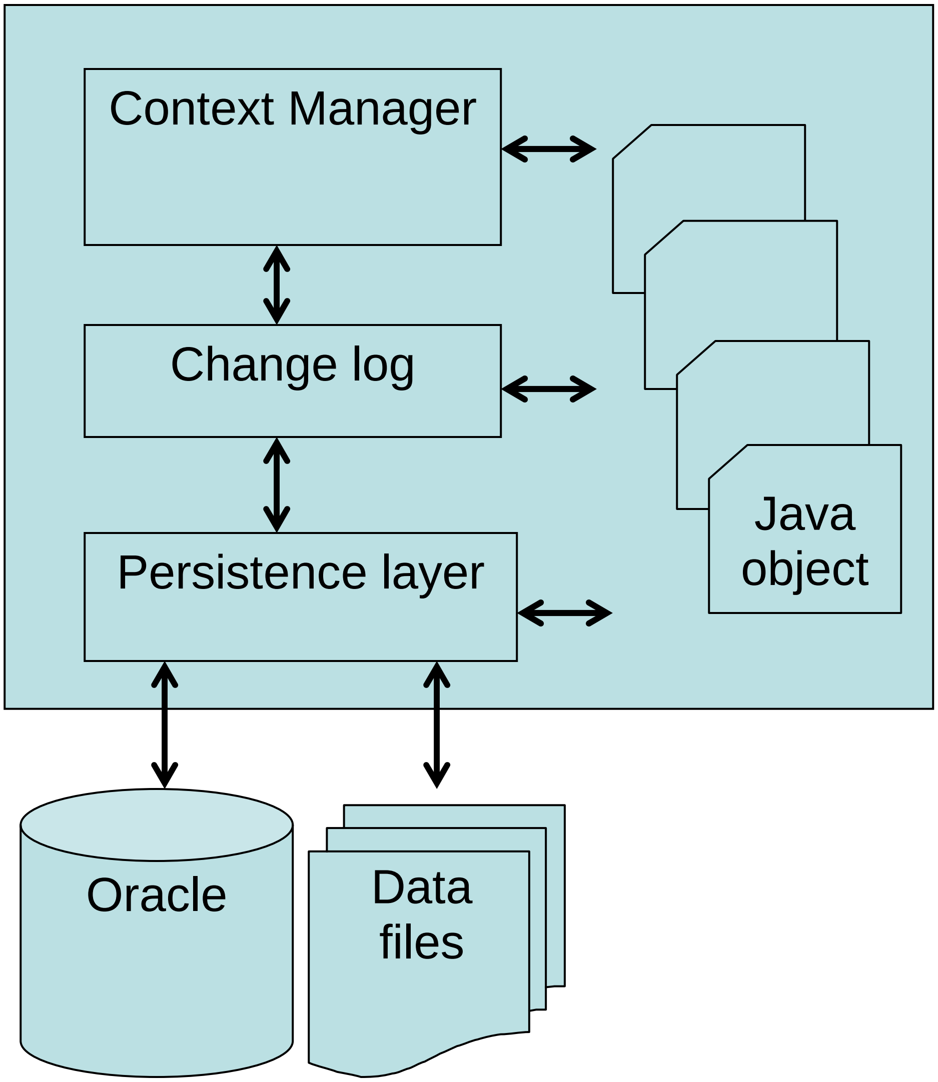
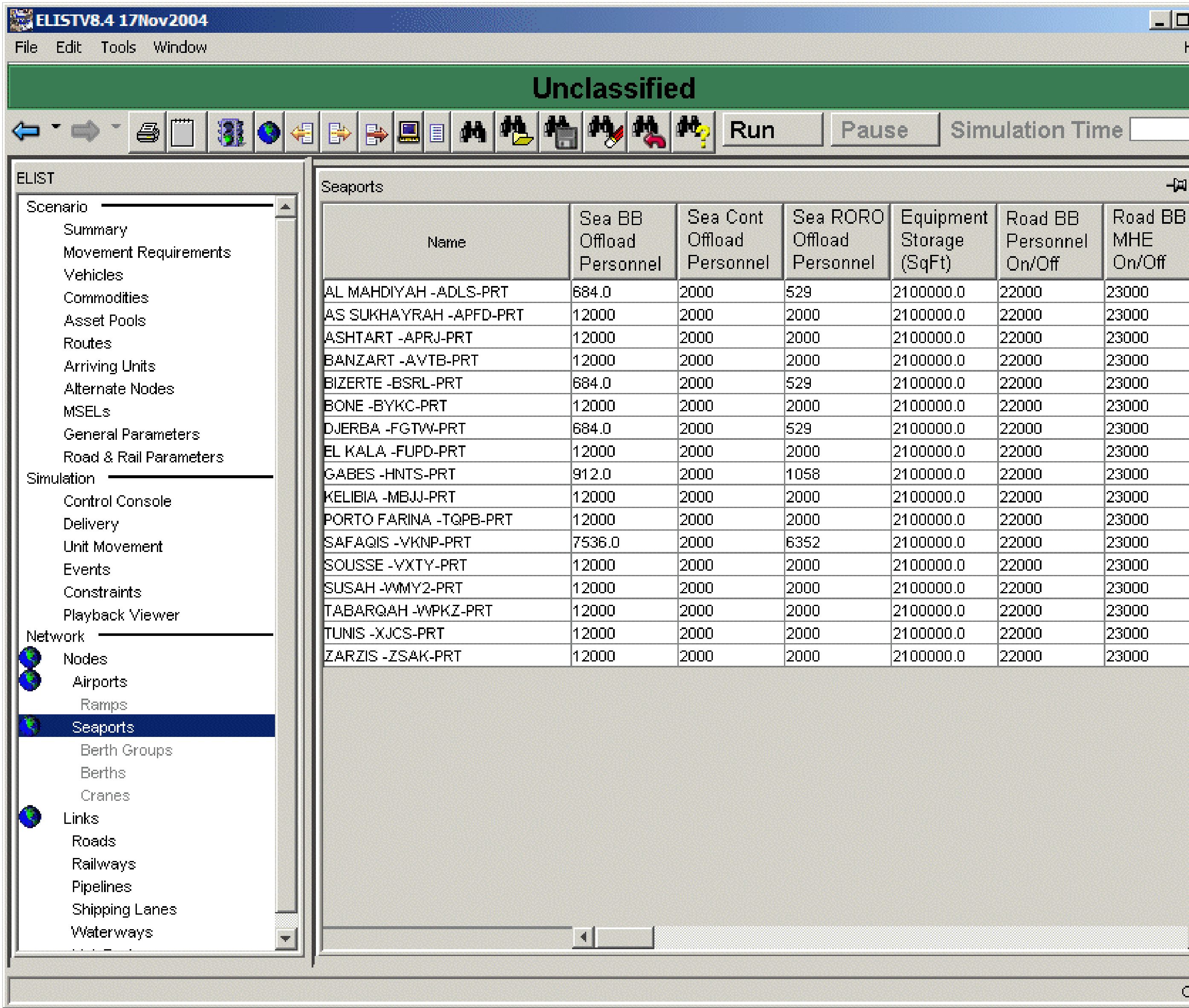


Figure 1: Structure of ELIST

Since ELIST is an analysis tool containing a discrete event simulation, users often want to examine or modify some portion of the scenario data during a simulation run. A context manager was designed to track the state of the system and control the data and functions available to the user in each state. States include: empty, scenario loaded (Figure 2), simulation running, and simulation paused (Figure 3). These states determine how data is displayed in the GUI. For instance, if the simulation has been run and the user is analyzing results, infrastructure and vehicle data are color coded to denote how much the resource was used during the simulation. If the user needs to change the value of the resource for a future run, he will not be forced to reset the simulation before the change is made, since he may want to continue to analyze the current run for more problems. To solve the dilemma of having edited data and simulation data no longer match – each changed value is denoted with a red pencil. Before the user can restart the simulation, the user must save changes (i.e., remove all the red pencils).



Name	Sea BB Offload Personnel	Sea Cont Offload Personnel	Sea RORO Offload Personnel	Equipment Storage (SqFt)	Road BB Personnel On/Off	Road BB MHE On/Off
AL MAHDIYAH -ADLS-PRT	684.0	2000	529	2100000.0	22000	23000
AS SUKHAYRAH -APFD-PRT	12000	2000	2000	2100000.0	22000	23000
ASHTART -APRJ-PRT	12000	2000	2000	2100000.0	22000	23000
BANZART -AVTB-PRT	12000	2000	2000	2100000.0	22000	23000
BIZERTE -BSRL-PRT	684.0	2000	529	2100000.0	22000	23000
BONE -BYKC-PRT	12000	2000	2000	2100000.0	22000	23000
DJERBA -FGTW-PRT	684.0	2000	529	2100000.0	22000	23000
EL KALA -FUPD-PRT	12000	2000	2000	2100000.0	22000	23000
GABES -HNTS-PRT	912.0	2000	1058	2100000.0	22000	23000
KELIBIA -MBJJ-PRT	12000	2000	2000	2100000.0	22000	23000
PORTO FARINA -TQPB-PRT	12000	2000	2000	2100000.0	22000	23000
SAFAQIS -VKNP-PRT	7536.0	2000	6352	2100000.0	22000	23000
SOUSSE -VXTY-PRT	12000	2000	2000	2100000.0	22000	23000
SUSAH -WMY2-PRT	12000	2000	2000	2100000.0	22000	23000
TABARQAH -WPKZ-PRT	12000	2000	2000	2100000.0	22000	23000
TUNIS -XJCS-PRT	12000	2000	2000	2100000.0	22000	23000
ZARZIS -ZSAK-PRT	12000	2000	2000	2100000.0	22000	23000

Figure 2: Network Editing in Scenario Edit Mode

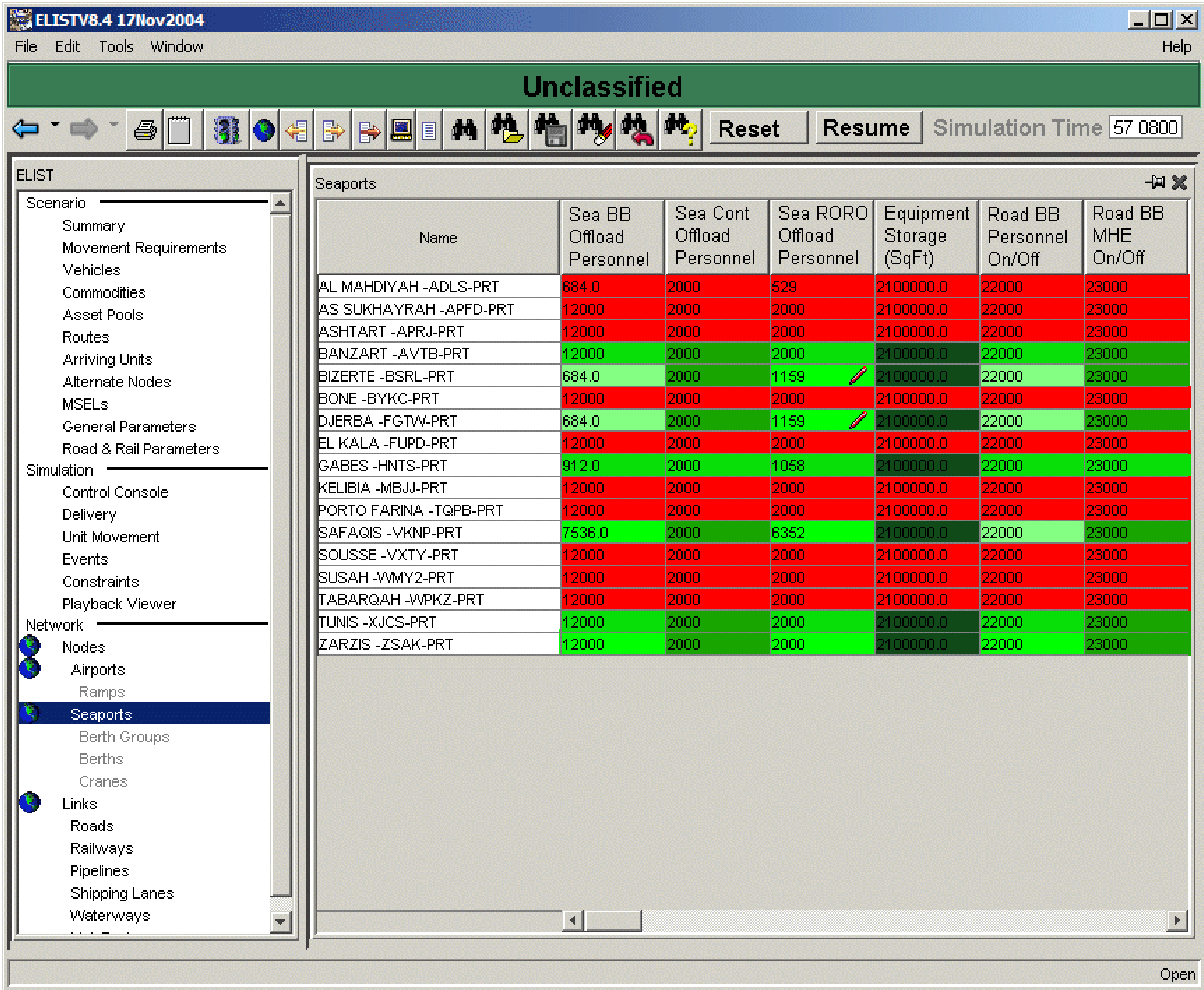


Figure 3: Network Editing in Simulation Mode

As the user changes data, a sequential log of changes is kept. This allows the user to undo changes made during the current session. When it is time to save the edits, the user has the option of “Save”, which applies the changes to the current data set or “Save As”, which copies the current data set, applies the changes to the copy, and sets the system to use the new data set.

When the user saves changes, the persistence layer saves the information to permanent storage [6]. The persistence layer consists of a set of objects that provide a mapping of data objects to relational tables and java object attributes to corresponding columns in the tables. When changes are saved to the database, the persistence layer creates the correct SQL insert, update, or delete calls to the database. This layered approach provides great flexibility for programmers, who focus on writing domain-specific java objects, while the saving and restoring is all performed by the persistence layer. The persistence layer is also responsible for loading all data. While the persistence layer was originally designed to work with relational databases, it is being expanded to also work with files. Each database table will be implemented by a comma separated file and each field with a column. The persistence layer will not be generating insert, update, and delete statements; but will instead recreate the files when save is called. This addition will help the program to be more flexible and the data more mobile.

3 Scenario Alternatives

ELIST has been designed to provide military planners the ability to compare the effects of many types of decisions in setting up the deployment of forces in theater. The military has many unique transportation requirements. For instance, non-secure item tracking has immense security implications as the enemy could track also movements. Moving individual vehicles throughout the theater increases their vulnerability, but moving in groups (called serials) increases the overhead of staging, tracking, and then providing security for each group. Additionally, the force is more vulnerable if it is not preceded by combat units which could secure the area. Finally, the reliability of transportation resources contracted from host-nation companies decreases during hostilities since many civilian drivers abandon their vehicles when danger presents itself.

Users can change many types of variables in a scenario such as:

- Asset Utilization: typically ELIST modelers tag about 10% of the vehicles to be unavailable due to maintenance. This number could be varied over time for contracted civilian workers based on the threat level.
- Transport Asset Allocation: given a set of assets, one could easily see how more or less of given assets affect deployment timelines. Users can also evaluate command and control issues by limiting when and where various companies can work in the theater. This approach goes beyond the “Do I have enough vehicles?” questions to also evaluate “Have I allocated the vehicles in the theater correctly?” questions.
- Tracking: it is tempting during a simulation to overlook the paperwork involved in moving forces. However, the expense of not doing so is often worse. Various functions have been included for tracking cargo and personnel to provide realistic evaluations of cutting corners. For instance, during the 1991 Gulf War, the military proved it could send lots of cargo there quickly. It also proved the military could lose track of a lot in storage areas. Evaluating the resources required to track cargo is essential, especially as logisticians try to reduce stockage levels and provide more just-in-time delivery.
- Storage: inadequate storage and staging facilities can reduce the flow of cargo dramatically. The more cluttered an area gets, the more work it takes to go around and reposition the items. The more times an item is handled the longer it takes to deploy and each item has a higher risk of damage. ELIST has shown that increased storage and staging areas can decrease the overall lateness of the delivery of forces in theater.
- Unit Phasing: while evaluating the deployment effects of putting forces in theater before transportation assets is relatively straightforward, it is not so easy to evaluate the risk of doing so. While risk is not specifically evaluated in ELIST, logisticians must consider both the delivery profiles and the risk profiles involved with phasing units. If port throughput units and transportation units arrive early, more combat units can be offloaded. But, if they come in before the area is stable, all forces may be lost.

In order to evaluate all of these factors, many variations of one basic scenario must be created and simulated. However, the amount of data used for an ELIST analysis is very large. Two methods for representing these alternatives efficiently have been developed. First, data has been separated into base data and scenario data. Base data refers to the very large and usually static network data and the very large movement requirement data. These data sets can be applied to many scenarios without needing to copy the data over and over. Scenario data is a combination of additional dynamic data (e.g. asset allocations and scenario assumptions) and references to base network and movement requirement data. Second, a scenario can include a set of changes to the base data which will be overlaid on the base data. This allows small alterations to be efficiently stored.

ELIST users are dispersed throughout diverse organizations of the military. They often need to share analyses, requiring efficient transfer of data sets. ELIST data is stored in 85 tables, each of which contains data from many scenarios. To send a data set to another site, users invoke the ELIST scenario export function. This process creates temporary Oracle™ tables populated with the data for the selected scenario, exports them to an Oracle™ dump file using the built-in Oracle™ utility exp, and then removes the temporary tables. When importing a scenario at a new location, new unique primary keys are required since the existing key may already be used in the local database. The ELIST import process uses the built-in Oracle™ imp utility to load the temporary tables from the dump file. It then updates the keys to be unique, copies the data into the scenario tables and then removes the temporary tables.

One of the unique features of this data management includes the option for the user to import parts of a scenario and reference existing base data sets. While this feature is fully implemented and available for users, in reality it is used infrequently. A major issue for the users is to ensure that the baseline data is consistent across machines. When copying data sets, users usually err on the side of data consistency rather on storage efficiency. A major lesson learned was that the software must balance efficiency and ease of use. The next release of ELIST will reduce the amount of available dependencies, only preserving those with very large data savings, as well as storing more information about the data (meta-data) so the users can be more confident about the contents of the data.

4 Evaluating Alternatives

To help evaluate a scenario, ELIST can generate hundreds of graphs and reports. There are some obvious graphs and reports to start evaluating the success or shortcomings of a given scenario. The contents of these initial reports and graphs point to others that need to be examined

First is the delivery graph, which shows whether items were delivered to their destinations and how much was early or late. Figure 4 provides a sample of how much cargo was delivered to the destinations vs. how much was required. While it looks good early on, there are some problems later. Figure 5 expands this information to show 1) if all items arrived to the theater on time (Theater Arrival), 2) if there was a delay offloading cargo from strategic transports (Offloading Delay), 3) if moving through the theater was a constraint (Theater Movement), and 4) if, even though the items are in theater, they are not usable because all of the unit has not arrived (Unit Closure).

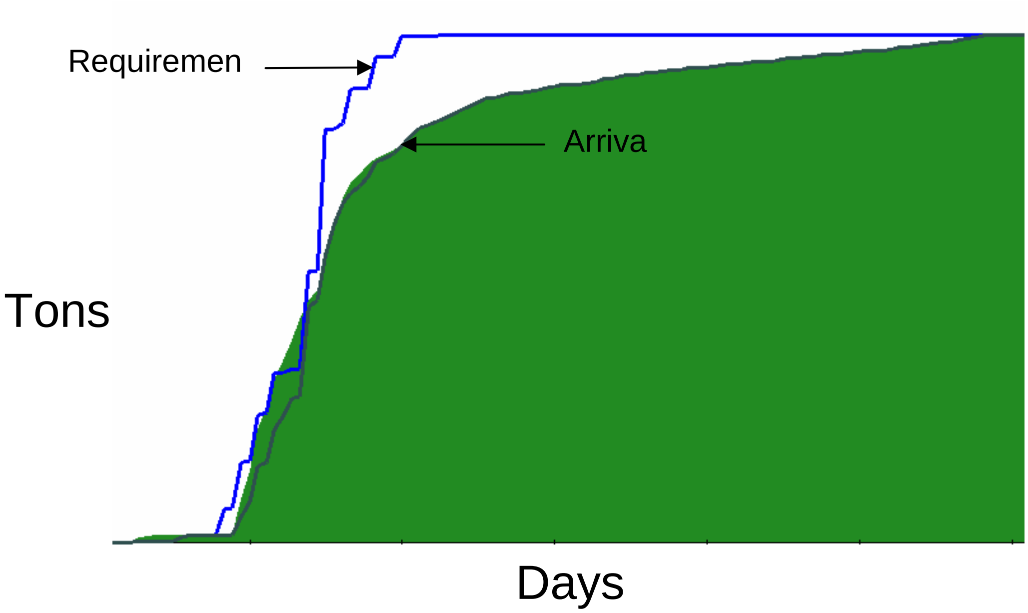


Figure 4: Sample Unit Arrival Graph

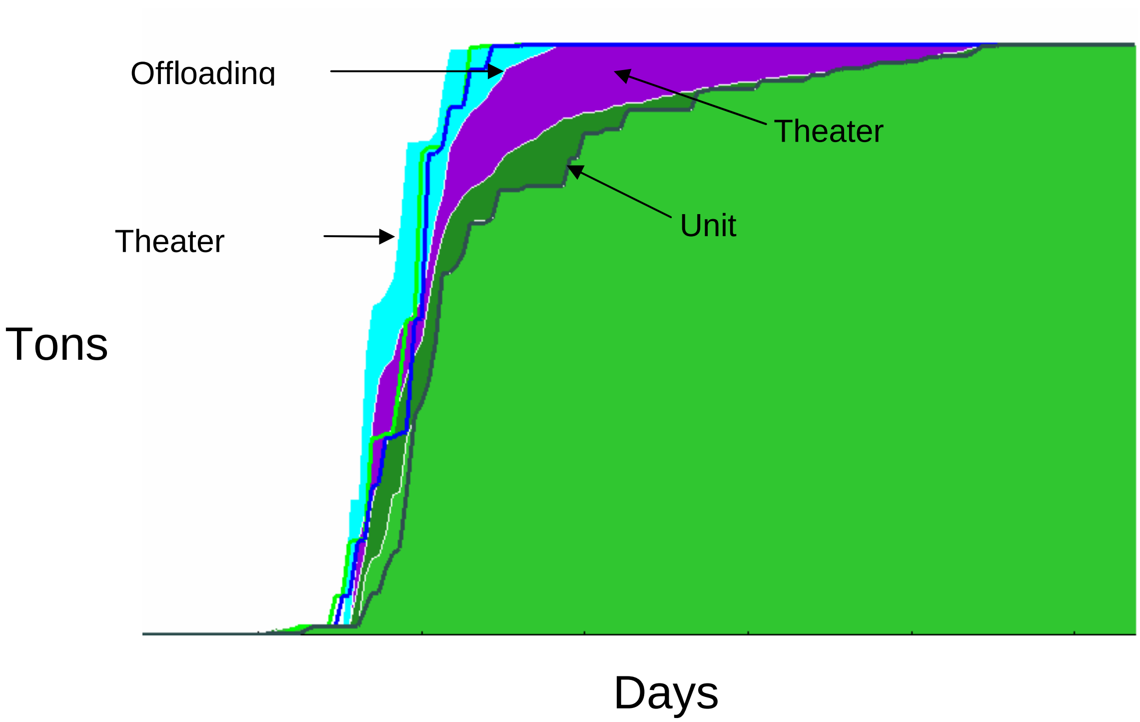
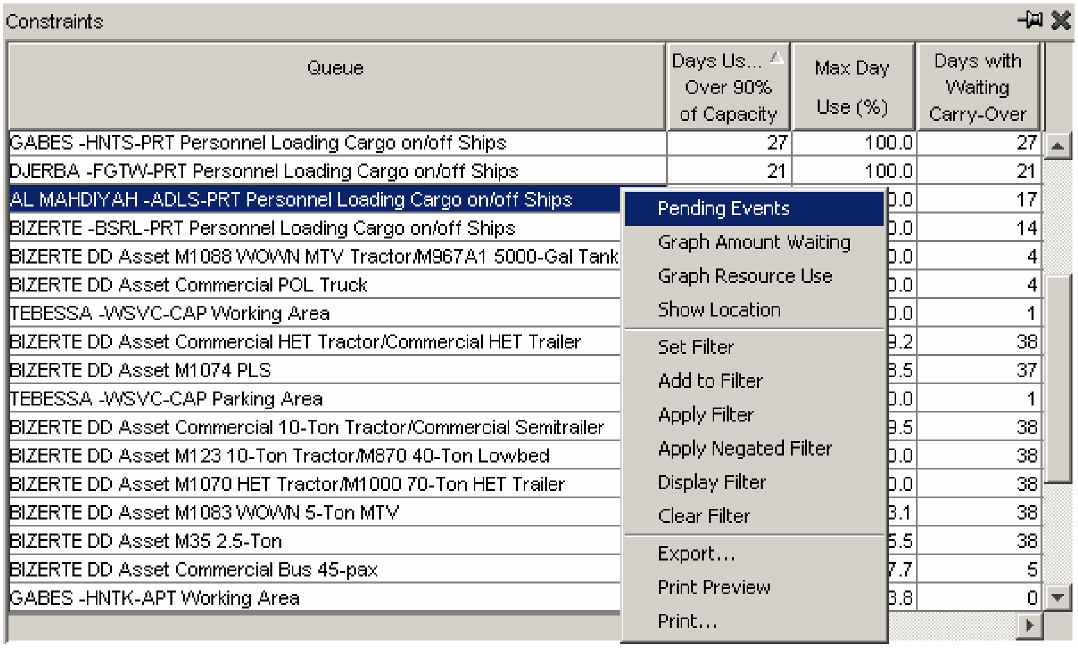


Figure 5: Sample Arrivals at Locations Graph

If there are problems, the users need to determine why there are delays. There are two overview reports that assist the users. The first is the bottlenecks report (Figure 6) which goes through each asset and resource in the simulation and determines if they were used at full capacity for extended periods. From these the user can directly graph the complete usage of the resource and check its utilization. As one can see, there are many alternatives for drilling down to find answers regarding why items either do or do not move through a theater. The second is the movement wait times report. This one is generated from a unit perspective, showing the delivery profile for each one.



Queue	Days Us... Over 90% of Capacity	Max Day Use (%)	Days with Waiting Carry-Over
GABES -HNTS-PRT Personnel Loading Cargo on/off Ships	27	100.0	27
DJERBA -FGTW-PRT Personnel Loading Cargo on/off Ships	21	100.0	21
AL MAHDIYAH -ADLS-PRT Personnel Loading Cargo on/off Ships		0.0	17
BIZERTE -BSRL-PRT Personnel Loading Cargo on/off Ships		0.0	14
BIZERTE DD Asset M1088 WOWN MTV Tractor/M967A1 5000-Gal Tank		0.0	4
BIZERTE DD Asset Commercial POL Truck		0.0	4
TEBESSA -WSVC-CAP Working Area		0.0	1
BIZERTE DD Asset Commercial HET Tractor/Commercial HET Trailer		9.2	38
BIZERTE DD Asset M1074 PLS		3.5	37
TEBESSA -WSVC-CAP Parking Area		0.0	1
BIZERTE DD Asset Commercial 10-Ton Tractor/Commercial Semitrailer		9.5	38
BIZERTE DD Asset M123 10-Ton Tractor/M870 40-Ton Lowbed		0.0	38
BIZERTE DD Asset M1070 HET Tractor/M1000 70-Ton HET Trailer		0.0	38
BIZERTE DD Asset M1083 WOWN 5-Ton MTV		3.1	38
BIZERTE DD Asset M35 2.5-Ton		5.5	38
BIZERTE DD Asset Commercial Bus 45-pax		7.7	5
GABES -HNTK-APT Working Area		3.8	0

Figure 6: Sample Constraints Report

A second level of analysis includes determining which of two scenarios is "better". Though there are many factors, the most obvious is to determine which scenario closed more units by their required delivery date. As was mentioned earlier, closure is only one aspect that must be considered. ELIST also provides tools for exporting data to other formats, like spreadsheets where data can be compared easily and flexibly.

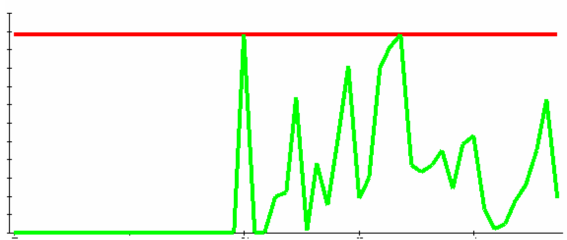


Figure 7: Sample Asset Usage Graph

5 Conclusions

ELIST provides users great flexibility to evaluate many policy and practice decisions that must be made by military planners each day. While most of the focus has been on evaluating individual scenarios, more work needs to be done to compare the results of multiple scenarios.

By generating tools to automatically save objects to the database or files, ELIST developers have been able to focus on modeling the problem, not coding input and output mechanisms. These tools also increase the reliability and maintainability of ELIST by centralizing GUI and saving mechanisms.

Data consistency between remotely installed versions of the software is a constant challenge. While earlier versions of ELIST focused on efficiently storing data, later versions have simplified this structure considerably because it could easily become confusing for users. In order to ensure data consistency, many users went ahead and stored duplicate data rather than risk having some data get out of sync. Additional information about the data itself has also been incorporated to assist the users in identifying differences in data sets.

Data consistency becomes even more difficult when data is shared with other models. ELIST has been incorporated into the Analysis of Mobility Platform [7], a federation of models that dynamically share data to simulate deployments from origin forts through tactical assembly areas in theater. In order to keep the models consistent, a common ontology must exist. This federation uses a common set of classes to pass all information between the models. While this ensures common objects, it does not fully cover common assumptions. More work is needed to ensure both models begin with the same assumptions and that decisions made in each model become known throughout the federation.

References

1. Van Groningen, C.N., Blachovicz, D., Braun, M.D., Widing, M.A., "Performing comprehensive reception, staging, onward movement, and integration analysis in a theater of operations", *Mathematical and Computer Modeling.* Vol. 39, No. 6 March 2004

2. M.D. Braun and C.N. Van Groningen, *ELIST 8 Transportation Model*, ANL/DIS/02-1, Argonne National Laboratory, Argonne, IL, (February 2002).

3. M. Braun, G. Lurie, K. Simunich, C. Van Groningen, H. Vander Zee and M. Widing, ELIST8: A simulation system for transportation logistics planning support, In *Proc. 2000 Summer Computer Simulation Conference*, Vancouver, BC, July 16{20, 2000.

4. Deployment Planning Guide, Transportation Assets Required for Deployment, MTMCTEA Reference 97-700-5, (July 1997).

5. Logistics Handbook for Strategic Mobility Planning, MTMCTEA Reference 97-700-2, (August 1997).

6. Ambler, Scott W. "The Design of a Robust Persistence Layer For Relational Databases"; available from http://www.ambysoft.com/persistenceLayer.pdf, November 28, 2000.

7. S. Stevens, J. Tustin and W. Key (2004). "U.S. Transportation Command Analysis of Mobility Platform Federation", European Simulation Interoperability Workshop 2004 (EuroSISO 2004).

About the Authors

Charles Van Groningen leads the ELIST development team at Argonne National Laboratory. He received a Ph.D. in Artificial Intelligence from the Illinois Institute of Technology in 1993, an M.S. in Computer Science from DePaul University, and a B.A. in Mathematics from Trinity Christian College. His research interests include modeling, simulation, and knowledge representation.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4832
Phone: (630) 252-5308
Fax: (630) 252-6073
vang@anl.gov

Dariusz Blachowicz is a Computer Scientist in the Decision and Information Science Division at Argonne National Laboratory. He received a B.S. in Civil Engineering and an M.S. in Computer Science from Illinois Institute of Technology in Chicago, Illinois. His work at Argonne includes a wide range of modeling and simulation applications, and web-based interactive applications for DOD and DOE agencies.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4832
Phone: (630) 252-6187
Fax: (630) 252-6073
blach@dis.anl.gov

Mary Braun is a Computer Systems Engineer in the Decision and Information Sciences Division at Argonne National Laboratory. Her work has focused on military logistics modeling and simulation. She has a B.S. from the University of Santa Clara and an M.S. from the University of California, Berkeley, both in Electrical Engineering.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4832
Phone: (630) 252-3727
Fax: (630) 252-6073
duffy@dis.anl.gov

Richard J. Love is a Software Engineer in the Decision and Information Sciences Division at Argonne National Laboratory. He earned an M.S. in computer science and a B.A. in physics from North Central College and a B.S. in electrical engineering from the University of Illinois. His experience and interests include graphical user interfaces, visualization, and Web-enabling technologies.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4832
Phone: (630) 252-9523
Fax: (630) 252-6073
love@dis.anl.gov

Kathy Lee Simunich is a Computer Scientist in the Decision and Information Sciences Division at Argonne National Laboratory. She received a B.S. in Meteorology from Northern Illinois University and an M.S. in Computer Science from North Central College in Illinois. Her work at Argonne includes environmental modeling and Object to Relational Data Bases, as well as writing reusable components across various DOD and DOE projects.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4832
Phone: (630) 252-3285

Fax: (630) 252-6073
simunich@dis.anl.gov

Mary Ann Widing is an Information Systems Engineer in the Decision and Information Sciences Division at Argonne National Laboratory. She received a B.S. and an M.S. in engineering from the University of Illinois in Urbana-Champaign. Her work at Argonne has focused on developing complex, graphical user interfaces for decision support systems used by government agencies.

Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4832
Phone: (630) 252-3798
Fax: (630) 252-6073
widing@dis.anl.gov

This work was supported under a military interdepartmental purchase request from the U.S. Department of Defense, Surface Deployment and Distribution Command Transportation Engineering Agency, through the U.S. Department of Energy contract W-31-109-ENG-109